МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИИ

федеральное государственное автономное образовательное учреждение высшего образования

Санкт-Петербургский национальный исследовательский университет информационных технологий механики и оптики

Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

**Лабораторная работа № 1**

**По дисциплине «Компьютерная геометрия и графика»**

**Изучение простых преобразований изображений.**

**Выполнила студентка группы №М3102***Абрамова Ульяна Владимировна*

**Проверил***Скаков Павел Сергеевич*

***САНКТ-ПЕТЕРБУРГ***

***2020***

1. цель работы

Изучить алгоритмы и реализовать программу, выполняющую простые преобразования серых и цветных изображений в формате PNM.

1. теоритическая часть
   1. Описание структуры PNM файлов

|  |  |
| --- | --- |
| PGM: Р5 ширина высота 255 Информация о пикселях (1 байт на пиксель) | PPM: Р6 ширина высота 255 Информация о пикселях (3 байта на пиксель) |

* 1. Описание преобразований

0 – ***инверсия*** – цвета становятся противоположными по спектру.

1 - ***зеркальное отражение по горизонтали*** – симметричное отображение относительно вертикальной оси симметрии.

2 - ***зеркальное отражение по вертикали*** – симметричное отображение относительно горизонтальной оси симметрии.

3 - ***поворот на 90 градусов по часовой стрелке***,

4 - ***поворот на 90 градусов против часовой стрелки***.

Преобразования исполняются следующим образом:

|  |  |  |
| --- | --- | --- |
| **преобразование** | **до** | **после** |
| *Инверсия* |  |  |
| *Отображение по горизонтали* |  |  |
| *Отображение по вертикали* |  |  |
| *Поворот по часовой* |  |  |
| *Поворот против часовой* |  |  |

1. Экспериментальная часть

Язык программирования: С++

Хранение данных

Данные считываются в массив arr типа char, размер которого равен width \* height + 1 в PGM или width \* height \* 3 + 1 в PPM (у меня нумерация в массиве идет с 1). Память под массив выделяется динамически. Считывание происходит при помощи функции read (buffer, size).

Инверсия

Каждое значение цвета отнимаем от 255 и записываем в файл.

Отображение по горизонтали

По строкам идем сверху вниз, а по столбцам справа налево и таким образом записываем в файл.

Отображение по вертикали

По строкам идем снизу вверх, а по столбцам справа налево и таким образом записываем в файл.

Поворот на 90 градусов по часовой стрелке

Идем по столбцам слева направо, а по строкам снизу вверх, и при записи в файл, меняем местами ширину и высоту.

Поворот на 90 градусов против часовой стрелки

Идем по столбцам справа налево, а по строкам сверху вниз, и при записи в файл меняем местами ширину и высоту.

Вывод

В ходе выполнения лабораторной работы, была разработана программа, способная выполнить простейшие указанные преобразования с изображениями форматов PGM и PPM.

Листинг кода

Файл main.cpp  
#include <iostream>  
#include <fstream>  
#include "cmake-build-debug/pgmwork.h"  
#include "cmake-build-debug/ppmwork.h"  
using namespace std;  
  
int width, height, a;  
string str;  
char\* arr;  
string filename\_in, filename\_out, command;  
  
void ItsPGM(ifstream &fin) {  
 arr = new char [height \* width + 1];  
 fin.read(arr, height \* width + 1);  
 if (fin)  
 cout << "all characters read successfully.";  
 else {  
 cout << "error: only " << fin.gcount() << " could be read";  
 return;  
 }  
 fin.close();  
 if(command == "0") Inverse(str, width, height, a, filename\_out, arr);  
 else if(command == "1") HorisonReflection(str, width, height, a, filename\_out, arr);  
 else if(command == "2") VerticalReflection(str, width, height, a, filename\_out, arr);  
 else if(command == "3") ClockwiseRotation(str, width, height, a, filename\_out, arr);  
 else if(command == "4") CounterclockwiseRotation(str, width, height, a, filename\_out, arr);  
 else cout << "Wrong command!\n";  
 delete[] arr;  
}  
  
void ItsPPM(ifstream &fin) {  
 arr = new char [height \* width \* 3 + 1];  
 fin.read(arr, height \* width \* 3 + 1);  
 if (fin)  
 std::cout << "all characters read successfully.";  
 else {  
 std::cout << "error: only " << fin.gcount() << " could be read";  
 return;  
 }  
 fin.close();  
 if(command == "0") Inverse1(str, width, height, a, filename\_out, arr);  
 else if(command == "1") HorisonReflection1(str, width, height, a, filename\_out, arr);  
 else if(command == "2") VerticalReflection1(str, width, height, a, filename\_out, arr);  
 else if(command == "3") ClockwiseRotation1(str, width, height, a, filename\_out, arr);  
 else if(command == "4") CounterclockwiseRotation1(str, width, height, a, filename\_out, arr);  
 else cout << "Wrong command!\n";  
 delete[] arr;  
}  
  
int main(int argc, char\* argv[])  
{  
 if(argc < 4){  
 cout << "Wrong arguments!\n";  
 return 1;  
 }  
 filename\_in = argv[1];  
 filename\_out = argv[2];  
 command = argv[3];  
 ifstream fin(filename\_in, ios::binary);  
 if (!fin) {  
 cout << "Cannot open file\n";  
 return 1;  
 }  
 if(fin.good())  
 fin >> str;  
 else{  
 cout << "Wrong\n";  
 exit(1);  
 }  
 if(fin.good())  
 fin >> width;  
 else {  
 cout << "Wrong\n";  
 exit(1);  
 }  
 if(fin.good())  
 fin >> height;  
 else {  
 cout << "Wrong\n";  
 exit(1);  
 }  
 if(fin.good())  
 fin >> a;  
 else {  
 cout << "Wrong\n";  
 exit(1);  
 }  
 if((width <= 0) || (height <= 0) || (a != 255)){  
 cout << "Something gone wrong!\n";  
 exit(1);  
 }  
 if(str == "P5")  
 ItsPGM(fin);  
 else if(str == "P6")  
 ItsPPM(fin);  
 else {  
 cout << "Wrong format!\n";  
 return 1;  
 }  
 fin.close();  
 return 0;  
}

Файл pgmwork.cpp  
#include "pgmwork.h"  
#include <fstream>  
using namespace std;  
  
void Inverse(string str, int width, int height, int a, string filename, char\* arr){  
 ofstream fout(filename, ios::binary);  
 if(!fout){  
 cout << "Cannot create file!\n";  
 return;  
 }  
 fout << str << "\n" << width << " " << height << "\n" << a << "\n";  
 for (int i = 1; i <= height \* width; i++)  
 fout << char(255 - (int)arr[i]);  
 fout.close();  
}  
  
void VerticalReflection(string str, int width, int height, int a, string filename, char\* arr) {  
 ofstream fout(filename, ios::binary);  
 if(!fout){  
 cout << "Cannot create file!\n";  
 return;  
 }  
 fout << str << "\n" << width << " " << height << "\n" << a << "\n";  
 for(int i = height - 1; i >= 0; i--)  
 for(int j = 1; j <= width; j++)  
 fout << arr[i \* width + j];  
 fout.close();  
}  
  
void HorisonReflection(string str, int width, int height, int a, string filename, char\* arr) {  
 ofstream fout(filename, ios::binary);  
 if(!fout){  
 cout << "Cannot create file!\n";  
 return;  
 }  
 fout << str << "\n" << width << " " << height << "\n" << a << "\n";  
 for(int i = 0; i < height; i++)  
 for(int j = width; j >= 1; j--)  
 fout << arr[i \* width + j];  
 fout.close();  
}  
  
void ClockwiseRotation(string str, int width, int height, int a, string filename, char\* arr){  
 ofstream fout(filename, ios::binary);  
 if(!fout){  
 cout << "Cannot create file!\n";  
 return;  
 }  
 fout << str << "\n" << height << " " << width << "\n" << a << "\n";  
 for(int i = 1; i <= width; i++)  
 for(int j = height - 1; j >= 0; j--)  
 fout << arr[j \* width + i];  
 fout.close();  
}  
  
void CounterclockwiseRotation(string str, int width, int height, int a, string filename, char\* arr){  
 ofstream fout(filename, ios::binary);  
 if(!fout){  
 cout << "Cannot create file!\n";  
 return;  
 }  
 fout << str << "\n" << height << " " << width << "\n" << a << "\n";  
 for(int i = width; i >= 1; i--)  
 for(int j = 0; j < height; j++)  
 fout << arr[j \* width + i];  
 fout.close();  
}

Файл ppmwork.cpp  
#include "ppmwork.h"  
#include <fstream>  
using namespace std;  
  
void Inverse1(string str, int width, int height, int a, string filename, char\* arr){  
 ofstream fout(filename, ios::binary);  
 if(!fout){  
 cout << "Cannot create file!\n";  
 return;  
 }  
 fout << str << "\n" << width << "\n" << height << "\n" << a << "\n";  
 for (int i = 1; i <= height \* width \* 3; i++)  
 fout << char(255 - (int)arr[i]);  
}  
  
void VerticalReflection1(string str, int width, int height, int a, string filename, char\* arr) {  
 ofstream fout(filename, ios::binary);  
 if(!fout){  
 cout << "Cannot create file!\n";  
 return;  
 }  
 fout << str << "\n" << width << " " << height << "\n" << a << "\n";  
 for(int i = height - 1; i >= 0; i--)  
 for(int j = 1; j <= width \* 3 - 2; j += 3)  
 fout << arr[i \* width \* 3 + j] << arr[i \* width \* 3 + j + 1] << arr[i \* width \* 3 + j + 2];  
 fout.close();  
}  
  
void HorisonReflection1(string str, int width, int height, int a, string filename, char\* arr) {  
 ofstream fout(filename, ios::binary);  
 if(!fout){  
 cout << "Cannot create file!\n";  
 return;  
 }  
 fout << str << "\n" << width << " " << height << "\n" << a << "\n";  
 for(int i = 0; i < height; i++)  
 for(int j = width \* 3; j >= 3; j -= 3)  
 fout << arr[i \* width \* 3 + j - 2] << arr[i \* width \* 3 + j - 1] << arr[i \* width \* 3 + j];  
 fout.close();  
}  
  
void ClockwiseRotation1(string str, int width, int height, int a, string filename, char\* arr){  
 ofstream fout(filename, ios::binary);  
 if(!fout){  
 cout << "Cannot create file!\n";  
 return;  
 }  
 fout << str << "\n" << height << " " << width << "\n" << a << "\n";  
 for(int i = 1; i < width \* 3 - 1 ; i += 3)  
 for(int j = height - 1; j >= 0; j--)  
 fout << arr[j \* width \* 3 + i] << arr[j \* width \* 3 + i + 1] << arr[j \* width \* 3 + i + 2];  
 fout.close();  
}  
  
void CounterclockwiseRotation1(string str, int width, int height, int a, string filename, char\* arr){  
 ofstream fout(filename, ios::binary);  
 if(!fout){  
 cout << "Cannot create file!\n";  
 return;  
 }  
 fout << str << "\n" << height << " " << width << "\n" << a << "\n";  
 for(int i = width \* 3; i >= 2; i -= 3)  
 for(int j = 0; j < height; j++)  
 fout << arr[j \* width \* 3 + i - 2] << arr[j \* width \* 3 + i - 1] << arr[j \* width \* 3 + i];  
 fout.close();  
}

Файл pgmwork.h  
#ifndef LAB1\_V3\_PGMWORK\_H  
#define LAB1\_V3\_PGMWORK\_H  
  
#include <string>  
#include <iostream>  
using namespace std;  
  
void Inverse(string str, int width, int height, int a, string filename, char\* arr); //0  
void HorisonReflection(string str, int width, int height, int a, string filename, char\* arr); //1  
void VerticalReflection(string str, int width, int height, int a, string filename, char\* arr); //2  
void ClockwiseRotation(string str, int width, int height, int a, string filename, char\* arr); //3  
void CounterclockwiseRotation(string str, int width, int height, int a, string filename, char\* arr); //4  
  
#endif //LAB1\_V3\_PGMWORK\_H

Файл ppmwork.h  
#ifndef LAB1\_V3\_PPMWORK\_H  
#define LAB1\_V3\_PPMWORK\_H  
  
#include <string>  
#include <iostream>  
using namespace std;  
  
void Inverse1(string str, int width, int height, int a, string filename, char\* arr); //0  
void HorisonReflection1(string str, int width, int height, int a, string filename, char\* arr); //1  
void VerticalReflection1(string str, int width, int height, int a, string filename, char\* arr); //2  
void ClockwiseRotation1(string str, int width, int height, int a, string filename, char\* arr); //3  
void CounterclockwiseRotation1(string str, int width, int height, int a, string filename, char\* arr); //4  
  
#endif //LAB1\_V3\_PPMWORK\_H